

Package ‘bnviewer’

August 21, 2019

Type Package

Title Interactive Visualization of Bayesian Networks

Version 0.1.4

Depends R (>= 3.4)

Author Robson Fernandes

Maintainer Robson Fernandes <robson.fernandes@usp.br>

Description Bayesian networks provide an intuitive framework for probabilistic reasoning and its graphical nature can be interpreted quite clearly. Graph based methods of machine learning are becoming more popular because they offer a richer model of knowledge that can be understood by a human in a graphical format. The 'bnviewer' is an R Package that allows the interactive visualization of Bayesian Networks. The aim of this package is to improve the Bayesian Networks visualization over the basic and static views offered by existing packages.

License MIT + file LICENSE

URL <http://robsonfernandes.net/bnviewer/>

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

Repository CRAN

Imports visNetwork (>= 2.0.4), bnlearn (>= 4.3), methods, igraph (>= 1.2.4)

NeedsCompilation no

Date/Publication 2019-08-21 07:40:07 UTC

R topics documented:

bnviewer-package	2
bn.to.igraph	2
model.to.structure	3
strength.viewer	4
viewer	8

bnviewer-package	<i>Interactive Visualization of Bayesian Networks</i>
------------------	---

Description

Bayesian networks provide an intuitive framework for probabilistic reasoning and its graphical nature can be interpreted quite clearly. Graph based methods of machine learning are becoming more popular because they offer a richer model of knowledge that can be understood by a human in a graphical format. The 'bnviewer' is an R Package that allows the interactive visualization of Bayesian Networks. The aim of this package is to improve the Bayesian Networks visualization over the basic and static views offered by existing packages.

Details

Package: bnviewer
Type: Package
Version: 0.1.4
Date: 2019-08-12
License: MIT + file LICENSE

Author(s)

Robson Fernandes
Institute of Mathematical and Computer Sciences
University of Sao Paulo - ICMC-USP
Maintainer: Robson Fernandes <robson.fernandes@usp.br>

bn.to.igraph	<i>Bayesian Network to iGraph Model</i>
--------------	---

Description

Converts Bayesian network structure based on package "bnlearn" and "bnviewer" to model based on package "igraph".

Usage

```
bn.to.igraph(bayesian.network)
```



```

model.right.arrow.op2 = " A => (B,C,D);
                        B => (E,F);
                        F => (G);
                        "

structure = model.to.structure(model.left.arrow.op1)

viewer(structure,
        bayesianNetwork.width = "100%",
        bayesianNetwork.height = "80vh",
        bayesianNetwork.layout = "layout_on_grid",
        node.colors = list(background = "#f4bafd",
                            border = "#2b7ce9",
                            highlight = list(background = "#97c2fc",
                                                border = "#2b7ce9"))
        )

data.set = bnlearn::gaussian.test

bayesianNetwork.fit = bn.fit(structure, data = data.set)

print(bayesianNetwork.fit$A)

```

strength.viewer

Interactive Bayesian Network Strength Viewer

Description

Show the strength of the probabilistic relationships expressed by the arcs of a Bayesian network, and use model averaging to build a network containing only the significant arcs.

Usage

```

strength.viewer(bayesianNetwork, bayesianNetwork.background = NULL,
                bayesianNetwork.boot.strength = NULL,
                bayesianNetwork.arc.strength.threshold.expression = NULL,
                bayesianNetwork.arc.strength.threshold.expression.color = NULL,
                bayesianNetwork.arc.strength.threshold.alternative.color = NULL,
                bayesianNetwork.arc.strength.label = FALSE,
                bayesianNetwork.arc.strength.label.prefix = "",
                bayesianNetwork.arc.strength.label.color = NULL,
                bayesianNetwork.arc.strength.tooltip = FALSE,
                bayesianNetwork.edge.scale.min = 1,
                bayesianNetwork.edge.scale.max = 5,
                bayesianNetwork.edge.scale.label.min = 14,
                bayesianNetwork.edge.scale.label.max = 14,
                bayesianNetwork.title = "", bayesianNetwork.subtitle = "",

```

```

bayesianNetwork.footer = "", bayesianNetwork.layout = "default",
bayesianNetwork.width = "100%", bayesianNetwork.height = "500px",
node.shape = c("dot"), node.label.prefix = "",
node.colors = list(), node.font = list(), edges.smooth = TRUE,
edges.dashes = FALSE, edges.colors = list(),
options.highlightNearest = TRUE, options.nodesIdSelection = FALSE)

```

Arguments

```

bayesianNetwork
    A Bayesian Network structure from Averaged Network
bayesianNetwork.background
    Bayesian network background
bayesianNetwork.boot.strength
    A nonparametric bootstrap to assess arc strength and direction
bayesianNetwork.arc.strength.threshold.expression
    Logical expression of the force threshold of the arcs of the Bayesian network
bayesianNetwork.arc.strength.threshold.expression.color
    Color applied to logical expression of the force threshold of the arcs of the
    Bayesian network
bayesianNetwork.arc.strength.threshold.alternative.color
    Alternative color to logical expression of the force threshold of the arcs of the
    Bayesian network
bayesianNetwork.arc.strength.label
    Enable Bayesian Network arc strength label
bayesianNetwork.arc.strength.label.prefix
    Include Bayesian Network arc strength label prefix
bayesianNetwork.arc.strength.label.color
    Set Bayesian Network arc strength label color
bayesianNetwork.arc.strength.tooltip
    Enable Bayesian Network arc strength tooltip
bayesianNetwork.edge.scale.min
    Set Bayesian Network edge scale minimum
bayesianNetwork.edge.scale.max
    Set Bayesian Network edge scale maximum
bayesianNetwork.edge.scale.label.min
    Set Bayesian Network edge scale label minimum
bayesianNetwork.edge.scale.label.max
    Set Bayesian Network edge scale label maximum
bayesianNetwork.title
    : String. Bayesian Network title
bayesianNetwork.subtitle
    : String. Bayesian Network subtitle
bayesianNetwork.footer
    : String. Bayesian Network footer

```

`bayesianNetwork.layout`
: String. A layout of a Bayesian Network

1. `layout_on_sphere`
2. `layout_on_grid`
3. `layout_in_circle`
4. `layout_as_star`
5. `layout_as_tree`
6. `layout_with_sugiyama`
7. `layout_with_kk`
8. `layout_with_dh`
9. `layout_with_lgl`
10. `layout_with_mds`
11. `layout_with_gem`
12. `layout_nicely`
13. `layout_components`
14. `layout_hierarchical_direction_UD`
15. `layout_hierarchical_direction_DU`
16. `layout_hierarchical_direction_LR`
17. `layout_hierarchical_direction_RL`

`bayesianNetwork.width`
: String. Bayesian Network width

`bayesianNetwork.height`
: String. Bayesian Network height

`node.shape` : String. A node shape of a Bayesian Network

1. `dot` (default)
2. `circle`
3. `ellipse`
4. `database`
5. `diamond`
6. `square`
7. `triangle`
8. `box`
9. `star`
10. `text`

`node.label.prefix`
: String. Adds a prefix to the node label

`node.colors` : String | named list. Color for the node. Can be `'rgba(120,32,14,1)'`, `'#97C2FC'` (hexa notation on 7 char without transparency) or `'red'`. Can be just one color, or a list with several elements:

1. `"background"` : String. Default to `'#97C2FC'`. Background color for the node.
2. `"border"` : String. Default to `'#2B7CE9'`. Border color for the node.

3. "highlight" : String | named list, Color of the node when selected.
 - (a) "background" : String. Default to '#97C2FC'. Background color for the node when selected.
 - (b) "border" : String. Default to '#2B7CE9'. Border color for the node when selected.

node.font	Node Font : Array. Example list(color = "black", face="Arial")
edges.smooth	: Boolean. When true, the edge is drawn as a dynamic quadratic bezier curve.
edges.dashes	: Array or Boolean. Default to false. When true, the edge will be drawn as a dashed line.
edges.colors	: Named list or String. Default to named list. Color information of the edge in every situation. Can be 'rgba(120,32,14,1)', '#97C2FC' (hexa notation on 7 char without transparency) or 'red'. <ul style="list-style-type: none"> • "color" : String. Default to '#848484'. The color of the edge when it is not selected or hovered over (assuming hover is enabled in the interaction module). • "highlight " : String. Default to '#848484'. The color the edge when it is selected. • "hover" : String. Default to '#848484'. The color the edge when the mouse hovers over it (assuming hover is enabled in the interaction module). • "inherit" : String or Boolean. Default to 'from'. When color, highlight or hover are defined, inherit is set to false! Supported options are: true, false, 'from','to','both'. • "opacity" : Number. Default to 1.0. It can be useful to set the opacity of an edge without manually changing all the colors. The allowed range of the opacity option is between 0 and 1.
options.highlightNearest	: Boolean. Default to true. Highlight nearest when clicking a node.
options.nodesIdSelection	: Boolean. Default to false. Add an id node selection creating an HTML select element.

References

See online documentation <http://robsonfernandes.net/bnviewer>

Examples

```
library(bnlearn)
library(bnviewer)

bayesianNetwork.boot.strength = boot.strength(coronary, R = 20, algorithm = "hc")

avg.bayesianNetwork = averaged.network(bayesianNetwork.boot.strength, threshold = 0.2)

strength.viewer(
  avg.bayesianNetwork,
```

```

bayesianNetwork.boot.strength,
bayesianNetwork.background = "white",
bayesianNetwork.arc.strength.threshold.expression = c("@threshold > 0 & @threshold < 0.5",
                                                    "@threshold >= 0.5 & @threshold < 0.6",
                                                    "@threshold >= 0.6 & @threshold <= 1"),

bayesianNetwork.arc.strength.threshold.expression.color = c("red", "yellow", "green"),
bayesianNetwork.arc.strength.threshold.alternative.color = "white",

bayesianNetwork.arc.strength.label = TRUE,
bayesianNetwork.arc.strength.label.prefix = "",
bayesianNetwork.arc.strength.label.color = "black",

bayesianNetwork.arc.strength.tooltip = TRUE,

bayesianNetwork.edge.scale.min = 1,
bayesianNetwork.edge.scale.max = 3,

bayesianNetwork.edge.scale.label.min = 14,
bayesianNetwork.edge.scale.label.max = 14,

bayesianNetwork.width = "100%",
bayesianNetwork.height = "800px",
bayesianNetwork.layout = "layout_with_sugiyama",
node.colors = list(background = "#97c2fc",
                   border = "#2b7ce9",
                   highlight = list(background = "#e91eba",
                                     border = "#2b7ce9")),

node.font = list(color = "black", face="Arial"),
edges.dashes = FALSE,

bayesianNetwork.title="Bayesian Network Strength Analysis - Coronary",
bayesianNetwork.subtitle = "Coronary heart disease data set",
bayesianNetwork.footer = "Fig. 1 - Layout with Sugiyama"
)

```

viewer

Interactive Bayesian Network Viewer

Description

Interactive Bayesian Network Viewer

Usage

```

viewer(bayesianNetwork, bayesianNetwork.background = NULL,
       bayesianNetwork.title = "", bayesianNetwork.subtitle = "",
       bayesianNetwork.footer = "", bayesianNetwork.layout = "default",

```



```

bayesianNetwork.width = "100%", bayesianNetwork.height = "500px",
node.shape = NULL, node.label.prefix = "", node.colors = list(),
node.font = list(), edges.smooth = TRUE, edges.dashes = FALSE,
options.highlightNearest = TRUE, options.nodesIdSelection = FALSE,
clusters.legend.title = "", clusters.legend.options = list(),
clusters = list()

```

Arguments

`bayesianNetwork`
A Bayesian Network structure. (Example : hill-climbing (HC)).

`bayesianNetwork.background`
Bayesian network background

`bayesianNetwork.title`
: String. Bayesian Network title

`bayesianNetwork.subtitle`
: String. Bayesian Network subtitle

`bayesianNetwork.footer`
: String. Bayesian Network footer

`bayesianNetwork.layout`
: String. A layout of a Bayesian Network. The hierarchical layout the available options are: UD, DU, LR, RL. To simplify: up-down, down-up, left-right, right-left.

1. `layout_on_sphere`
2. `layout_on_grid`
3. `layout_in_circle`
4. `layout_as_star`
5. `layout_as_tree`
6. `layout_with_sugiyama`
7. `layout_with_kk`
8. `layout_with_dh`
9. `layout_with_lgl`
10. `layout_with_mds`
11. `layout_with_gem`
12. `layout_nicely`
13. `layout_components`
14. `layout_hierarchical_direction_UD`
15. `layout_hierarchical_direction_DU`
16. `layout_hierarchical_direction_LR`
17. `layout_hierarchical_direction_RL`

`bayesianNetwork.width`
: String. Bayesian Network width

`bayesianNetwork.height`
: String. Bayesian Network height

<code>node.shape</code>	: String. A node shape of a Bayesian Network <ol style="list-style-type: none"> 1. dot (default) 2. circle 3. ellipse 4. database 5. diamond 6. square 7. triangle 8. box 9. star 10. text
<code>node.label.prefix</code>	: String. Adds a prefix to the node label
<code>node.colors</code>	: String named list. Color for the node. Can be <code>'rgba(120,32,14,1)'</code> , <code>'#97C2FC'</code> (hexa notation on 7 char without transparency) or <code>'red'</code> . Can be just one color, or a list with several elements: <ol style="list-style-type: none"> 1. "background" : String. Default to <code>'#97C2FC'</code>. Background color for the node. 2. "border" : String. Default to <code>'#2B7CE9'</code>. Border color for the node. 3. "highlight" : String named list, Color of the node when selected. <ol style="list-style-type: none"> (a) "background" : String. Default to <code>'#97C2FC'</code>. Background color for the node when selected. (b) "border" : String. Default to <code>'#2B7CE9'</code>. Border color for the node when selected.
<code>node.font</code>	Node Font : Array. Example <code>list(color = "black", face="Arial")</code>
<code>edges.smooth</code>	: Boolean. When true, the edge is drawn as a dynamic quadratic bezier curve.
<code>edges.dashes</code>	: Array or Boolean. Default to false. When true, the edge will be drawn as a dashed line.
<code>options.highlightNearest</code>	: Boolean. Default to true. Highlight nearest when clicking a node.
<code>options.nodesIdSelection</code>	: Boolean. Default to false. Add an id node selection creating an HTML select element.
<code>clusters.legend.title</code>	: Array. Get details in the example.
<code>clusters.legend.options</code>	: Array of Array. Get details in the example.
<code>clusters</code>	: Array of Array. Get details in the example.

References

See online documentation <http://robsonfernandes.net/bnviewer>

See the code fontAwesome for icons in groups and nodes <https://fontawesome.com/v4.7.0/cheatsheet/>


```
clusters = list(  
  list(label = "Pressure",  
        shape = "icon",  
        icon = list(code = "f1ce", color = "#e91e63"),  
        nodes = list("CVP", "BP", "HRBP", "PAP", "PRSS")),  
  list(label = "Volume",  
        shape = "icon",  
        icon = list(code = "f140", color = "#03a9f4"),  
        nodes = list("MINV", "MVS", "LVV", "STKV")),  
  list(label = "Ventilation",  
        shape = "icon",  
        icon = list(code = "f192", color = "#4caf50"),  
        nodes = list("VALV", "VLNG", "VTUB", "VMCH")),  
  list(label = "Saturation",  
        shape = "icon",  
        icon = list(code = "f10c", color = "#ffc107"),  
        nodes = list("HRSA", "SAO2", "PVS"))  
)  
)
```

Index

`bn.to.igraph`, [2](#)
`bnviewer` (`bnviewer-package`), [2](#)
`bnviewer-package`, [2](#)
`model.to.structure`, [3](#)
`strength.viewer`, [4](#)
`viewer`, [8](#)